

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.

#^{RS}
2

Docket No.: P-202

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of

Dae Sik KIM

Serial No.: New U.S. Patent Application

Filed: March 9, 2001

For: INTERLEAVER MEMORY ACCESS APPARATUS AND METHOD
OF CDMA SYSTEM

J1036 U.S. PTO
09/801637
03/09/01

TRANSMITTAL OF CERTIFIED PRIORITY DOCUMENT

Assistant Commissioner of Patents
Washington, D. C. 20231

Sir:

At the time the above application was filed, priority was claimed based on the
following application:

Korean Patent Application No. 12257/2000 filed March 11, 2000.

A copy of each priority application listed above is enclosed.

Respectfully submitted,
FLESHNER & KIM, LLP

Daniel Y.J. Kim
Registration No. 36,186

P. O. Box 221200
Chantilly, Virginia 20153-1200
703 502-9440

Date: March 9, 2001

DYK/kam

대한민국 특허청

KOREAN INDUSTRIAL
PROPERTY OFFICE

J1036 U.S. PTO
09/801637



별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto
is a true copy from the records of the Korean Industrial
Property Office.

출원번호 : 특허출원 2000년 제 12257 호
Application Number

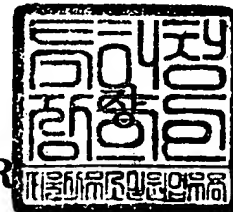
출원년월일 : 2000년 03월 11일
Date of Application

출원인 : 엘지정보통신주식회사
Applicant(s)

2000 년 10 월 09 일

특 허 청

COMMISSIONER



【서류명】	특허출원서		
【권리구분】	특허		
【수신처】	특허청장		
【참조번호】	0007		
【제출일자】	2000.03.11		
【국제특허분류】	H04B		
【발명의 명칭】	인터리버 메모리 제어 장치 및 방법		
【발명의 영문명칭】	Interleaver memory control apparatus and method		
【출원인】			
【명칭】	엘지정보통신 주식회사		
【출원인코드】	1-1998-000286-1		
【대리인】			
【성명】	강용복		
【대리인코드】	9-1998-000048-4		
【포괄위임등록번호】	1999-057037-3		
【대리인】			
【성명】	김용인		
【대리인코드】	9-1998-000022-1		
【포괄위임등록번호】	1999-057038-1		
【발명자】			
【성명의 국문표기】	김대식		
【성명의 영문표기】	KIM,Dae Sik		
【주민등록번호】	660727-1240721		
【우편번호】	435-042		
【주소】	경기도 군포시 산본2동 1066번지 개나리아파트 1334-402		
【국적】	KR		
【심사청구】	청구		
【취지】	특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사를 청구합니다. 대리인 강용복 (인) 대리인 김용인 (인)		
【수수료】			
【기본출원료】	20	면	29,000 원
【가산출원료】	1	면	1,000 원

1020000012257

2000/10/1

【우선권주장료】	0	건	0	원
【심사청구료】	3	항	205,000	원
【합계】	235,000			원
【첨부서류】	1.	요약서·명세서(도면)_1통		

【요약서】**【요약】**

본 발명은 인터리버 메모리 제어 장치 및 방법에 관한 것으로 특히 코드분할 다중 접속 이동통신 시스템의 송신 데이터 처리시 사용되는 인터리버 메모리의 어드레싱 및 액세스 방법을 개선하여 메모리 액세스에 사용되는 전력 소모의 감소 및 데이터 액세스 속도를 증가시킬 수 있는 인터리버 메모리 제어 장치 및 방법에 관한 것이다. 이와 같은 인터리버 메모리 제어 장치는 복수의 행(row)과, 복수의 열(column) 방향으로 코드 심볼 데이터를 저장하는 블록 인터리버 메모리부와, 상기 블록 인터리버 메모리부에서의 하나의 행(row) 방향의 코드 심볼 데이터를 순차적으로 전송받는 메모리부와, 상기 메모리부의 상기 복수개의 코드 심볼 데이터를 분할하여 수신하고, 수신된 코드 심볼 데이터를 64-ary 직교 변조를 위한 인덱스 디코딩을 실시하는 복수개의 인덱스 디코더와, 상기 블록 인터리버 메모리부로 한번에 하나의 행 방향의 모든 데이터를 전송하도록 제어신호를 발생하고, 상기 메모리부 및 복수개의 인덱스 디코더를 제어하는 제어 신호를 발생시키는 어드레스 발생 및 로직 제어부로 구성된다.

【대표도】

도 4

【색인어】

인터리버 메모리 제어

【명세서】**【발명의 명칭】**

인터리버 메모리 제어 장치 및 방법{Interleaver memory control apparatus and method}

【도면의 간단한 설명】

도 1은 일반적인 코드분할 다중접속 시스템의 블록 구성도

도 2는 종래 인터리버 메모리 맵을 나타낸 도면

도 3은 종래 메모리 데이터의 리드 어드레싱을 설명하기 위한 도면

도 4는 본 발명에 따른 인터리버 메모리 및 제어 로직을 설명하기 위한 블록 구성도

도 5는 본 발명에 따른 인터리버 메모리의 동작을 설명하기 위한 도면

도 6은 본 발명 제 1 실시예에 따른 인터리버 메모리의 리드 어드레싱 및 해당 데이터의 출력값을 나타낸 도면

도 7은 본 발명 제 2 실시예에 따른 인터리버 메모리의 리드 어드레싱 및 해당 데이터의 출력값을 나타낸 도면

도 8은 본 발명 제 3 실시예에 따른 인터리버 메모리의 리드 어드레싱 및 해당 데이터의 출력값을 나타낸 도면

도 9는 본 발명 제 4 실시예에 따른 인터리버 메모리의 리드 어드레싱 및 해당 데이터의 출력값을 나타낸 도면

도면의 주요 부분에 대한 부호의 설명

10 : 어드레스 발생 및 제어 로직부 11 : 블록 인터리버 메모리부

12 : 쉬프트 레지스터 13 : 인덱스 디코더부

14 : 64-ary 직교 변조기

【발명의 상세한 설명】

【발명의 목적】

【발명이 속하는 기술분야 및 그 분야의 종래기술】

<14> 본 발명은 인터리버 메모리 제어에 관한 것으로 특히 코드분할 다중접속 이동통신 시스템의 송신 데이터 처리시 사용되는 인터리버 메모리의 어드레싱 및 액세스 방법을 개선하여 메모리 액세스에 사용되는 전력 소모의 감소 및 데이터 액세스 속도를 증가시킬 수 있는 인터리버 메모리 제어 장치 및 방법에 관한 것이다.

<15> 이하, 첨부된 도면을 참조하여 일반적인 코드분할 다중접속 시스템의 인터리버 메모리 데이터의 리드 어드레싱을 설명하기로 한다.

<16> 도 1은 일반적인 코드분할 다중접속 시스템의 블록 구성도이다.

<17> 일반적인 코드분할다중접속(이하 CDMA라 약칭 함) 시스템은 아날로그 음성입력이 펄스 코드 변조(PCM)되고, 보코더(Vocoder)되어 프레임 품질 지시자(1)로 입력되면, 엔코더 테일 8비트 추가(2)후 길쌈 부호화(Convolutional Encoder)(R=1/3,K=9)(역방향 통화채널의 경우)(1개의 비트 입력에 대해 3개의 심볼이 생기므로 9.6Kbps의 입력에 대해 28.8Kbps의 출력이 생긴)(3)한다. 이어서 코드 심볼 반복기(Symbol repetition)(4), 블록 인터리버(Block Interleaver)(5)를 통해 64-ary 직교 변조기(6)에서 블록 인터리버(5)를 거친 28.8Kbps의 출력을 6심

불씩 디코딩하여 하나의 인덱스를 이용하여 64개의 월시(Walsh) 코드중 하나를 선택하여 6심볼 대신 선택한 월시 코드를 전송한다. 이때, 6비트로 지정할 수 있는 가지수는 $2^6=64$ 이다. 그리고, 64-ary 직교 변조기(6)에서 6심볼씩 자르면 $28.8/6=4.8\text{Kbps}$ 가 되고, 이것들이 64chip들을 만들어 내므로 $28.8/6*64=307.2\text{Kcps}$ 가 된다. 이어서 데이터 버스트 랜덤화기(7) 및 변조기(8)를 거쳐 데이터가 전송된다.

<18> 도 2는 종래 인터리버 메모리 맵을 나타낸 도면이고, 도 3은 종래 메모리 데이터의 리드 어드레싱을 설명하기 위한 도면이다.

<19> 종래 인터리버 메모리 맵은 인터리버 메모리의 데이터 포지션(position) 및 어드레싱(addressing)에 대한 구성을 나타낸 것으로, CDMA 시스템에서는 가변 데이터 레이트(variable data rate)를 지원하고 있고, 각각의 데이터 레이트에 따라 구성된 데이터의 특성을 설명하기 위한 것이다.

<20> 종래 데이터 레이트는 풀 레이트(Full rate), 하프 레이트(Half rate), 쿼터(Quarter rate) 및 에이트 레이트(Eight rate)가 있는데, 데이터 처리의 용이성을 위하여 풀 레이트를 제외한 데이터 레이트(하프, 쿼터 및 에이트 레이트)에 대하여는 코드 심볼 반복을 실시하여 풀 레이트와 동일한 데이터 사이즈로 만든다.

<21> 이동통신 시스템에서는 무선(air) 인터페이스를 통하여 데이터 전송이 이루어지고, 따라서 각종 노이즈(noise)에 대하여 예기치 않은 데이터의 손실이 발생할 수 있다. 이러한 경우에 디지털 통신 시스템의 특성상 버스트(burst) 에러가 아닌 경우에는 몇 개의 에러에 대해서는 교정(correction) 능력을 가진다. 이렇게 전송데이터의 버스트 에러를 방지하기 위한 보편적인 방법이 인터리빙(interleaving) 기술이다.

- <22> 종래에는 이러한 인터리빙 기술이 심플하게 메모리 뱅크와 일정한 패턴으로 발생 (generation)하는 어드레스 발생기를 구성하여 심볼 반복기로부터 전송되는 시리얼 데이터를 도 3에 나타낸 종래 인터리버 맵의 위치에 1부터 576까지의 1프레임 데이터를 라이트(write) 한다. 이때의 라이트 어드레스는 Row(1 ~ 32), Column(1 ~ 18)= 576까지 연속적으로 발생한다.
- <23> 이와 같은 라이트 동작 후에 데이터 리드 아웃 동작은 CDMA 통신 규약에서 정하는 순서대로 액세스 하는데 이 방법은 인터리버 메모리의 Row(1 ~ 32)순서대로 진행된다.
- <24> 규약에서 정하는 액세스 Row의 순서는 다음과 같다.
- <25> 풀 레이트(Full rate) : 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19, 20,21,22,23,24,25,26,27,28,29,30,31,32.
- <26> 하프 레이트(Half rate) : 1,3,2,4,5,7,6,8,9,11,10,12,13,15,14,16,17,19, 18,20,21,23,22,24,25,27,26,28,29,31,30,32.
- <27> 쿼터(Quarter rate):1,5,2,6,3,7,4,8,9,13,10,14,11,15,12,16,17,21,18,22, 19,23,20,24,25,29,26,30,27,31,28,32.
- <28> 에이트 레이트(Eight rate) :1,9,2,10,3,11,4,12,5,13,6,14,7,15,8,16,17, 25,18,26,9,7,20,28,21,29,22,30,23,31,24,32.
- <29> 즉 데이터 라이트 동작은 시리얼 입력의 순서대로(데이터의 연속성을 유지) 1~ 576까지 라이트하고, 데이터 리드는 정해진 Row 단위로 액세스함으로써 데이터의 연속성을 지향하면서 데이터의 버스트 에러에 대비하기 위한 데이터 스프레드(spread)가 이뤄진다.

<30> 이와 같은 종래 인터리버 메모리 제어에 있어서는 인터리버 메모리의 동작이 데이터 비트 단위로 연속적으로 이루어지므로 매 데이터 프레임(20ms)마다 인터리버 메모리의 동작을 반복하고 이로 인해 메모리 액세스 및 어드레스 발생이 매번 이루어지므로 이동 통신 시스템에서 요구되는 신속한 데이터 처리 및 전력 소모의 최소화에 많은 제약이 되고 특히 휴대폰의 경우에 배터리에서 전력 손실이 발생하는 문제점이 있었다.

【발명이 이루고자 하는 기술적 과제】

<31> 본 발명의 목적은 이상에서 언급한 종래 기술의 문제점을 감안하여 안출한 것으로서, 송신 데이터 처리시 사용되는 인터리버 메모리의 어드레싱 및 액세스 방법을 개선하여 메모리 액세스에 사용되는 전력 소모의 감소 및 데이터 액세스 속도를 증가시킬 수 있는 인터리버 메모리 제어 장치 및 방법을 제공하기 위한 것이다.

<32> 이상과 같은 목적을 달성하기 위한 본 발명의 일 특징에 따르면, 복수의 행(row)과, 복수의 열(column) 방향으로 코드 심볼 데이터를 저장하는 블록 인터리버 메모리부와, 상기 블록 인터리버 메모리부에서의 하나의 행(row) 방향의 코드 심볼 데이터를 순차적으로 전송받는 메모리부와, 상기 메모리부의 상기 복수개의 코드 심볼 데이터를 분할하여 수신하고, 수신된 코드 심볼 데이터를 64-ary 직교 변조를 위한 인덱스 디코딩을 실시하는 복수개의 인덱스 디코더와, 상기 블록 인터리버 메모리부로 한번에 하나의 행 방향의 모든 데이터를 전송하도록 제어신호를 발생하고, 상기 메모리부 및 복수개의 인덱스 디코더를 제어하는 제어 신호를 발생시키는 어드레스 발생 및 로직 제어부로 구성된다.

<33> 바람직하게, 복수개의 인덱스 디코더는 3개로 구성된다.

<34> 이상과 같은 다른 목적을 달성하기 위한 본 발명의 다른 특징에 따르면, 블록 인터리버 메모리부에서 설정된 비트(bit) 수만큼 복수개의 행(row)과, 복수개의 열(column) 방향으로 코드 심볼 데이터를 저장하는 단계와, 상기 블록 인터리버 메모리부에서 한 클럭(clock)마다 한 개씩의 상기 행(row) 방향의 모든 코드 심볼 데이터를 메모리부로 전송하는 단계와, 상기 메모리부에서 복수개의 인덱스 디코더로 상기 하나의 행 방향의 코드 심볼 데이터를 상기 인덱스 디코더의 개수로 분할하여 전송하는 단계와, 상기 복수개의 인덱스 디코더의 각각에서 상기 전송된 심볼 데이터를 인덱스 디코딩하는 단계로 이루어진다.

【발명의 구성 및 작용】

<35> 이하 본 발명의 바람직한 일 실시 예에 따른 구성 및 작용을 첨부된 도면을 참조하여 설명한다.

<36> 도 4는 본 발명에 따른 인터리버 메모리 및 제어 로직을 설명하기 위한 블록 구성도이고, 도 5는 본 발명에 따른 인터리버 메모리의 동작을 설명하기 위한 도면이며, 도 6은 본 발명 제 1 실시예에 따른 인터리버 메모리의 리드 어드레싱 및 해당 데이터의 출력값을 나타낸 도면이고, 도 7은 본 발명 제 2 실시예에 따른 인터리버 메모리의 리드 어드레싱 및 해당 데이터의 출력값을 나타낸 도면이며, 도 8은 본 발명 제 3 실시예에 따른 인터리버 메모리의 리드 어드레싱 및 해당 데이터의 출력값을 나타낸 도면이고, 도 9는 본 발명 제 4 실시예에 따른 인터리버 메모리의 리드 어드레싱 및 해당 데이터의 출력값을 나타낸 도면이다.

<37> 본 발명에 따른 인터리버 메모리 및 제어 로직은 행(row)과 열(column) 방향으로 코드 심볼 데이터를 저장하는 블록 인터리버 메모리부(11)와, 상기 블록 인터리버 메모리

리부(11)에서의 하나의 행(row) 방향의 코드 심볼 데이터를 순차적으로 전송받는 메모리인 쉬프트 레지스터(12)와, 상기 쉬프트 레지스터(12)에 전송된 복수개의 코드 심볼 데이터를 64-ary 직교 변조를 위한 인덱스 디코딩을 실시하는 복수개의 인덱스 디코더(13a, 13b, 13c)와, 상기 블록 인터리버 메모리부(11)로 인터리버 메모리 액세스를 위한 메모리 액세스 어드레스를 발생시키고, 쉬프트 레지스터(12) 및 복수개의 인덱스 디코더(13a, 13b, 13c)를 제어하는 제어 신호를 발생시키는 어드레스 발생 및 로직 제어부(10)로 구성된다.

<38> 이와 같은 본 발명 인터리버 메모리 제어 장치는 일정 사이즈(예를 들면, 576비트)의 메모리인 블록 인터리버 메모리부(11)에 코드 심볼 반복기(도 1의 4)에서 전송되는 심볼 코드가 열(column) 방향으로 순차적으로 라이트(write)된다. 이때, 행(Row)으로는 1 ~ 32비트의 심볼 코드가 열(column)로는 1 ~ 18비트의 심볼코드가 라이트 된다. 이는 도 5에 나타낸 바와 같은 인터리버 메모리의 동작을 설명하기 위한 도면을 보면 이해할 수 있을 것이다.

<39> 이어 블록 인터리버 메모리부(11)의 행(row) 및 열(column)로 설정된 메모리만큼의 심볼 코드가 전부 라이트되면, 어드레스 발생 및 로직 제어부(10)는 1번째에서부터 32번째 행(row)방향의 코드 심볼 데이터(1 ~ 32-Row Data)를 한번에 한 행(row)씩 순차적으로 쉬프트 레지스터(12)로 전송하도록 제어한다. 즉, 한번에 18비트의 행(row) 데이터를 전송한다.

<40> 이어서, 쉬프트 레지스터(12)는 블록 인터리버 메모리부(11)에서 전송된 18비트의 데이터를 어드레스 발생 및 제어 로직부(10)의 데이터 셀렉트 신호에 따라 1클럭 사이클(clock cycle)에 18비트의 데이터를 쉬프트 레지스터로 전송 후 제어로직부(10)의 데

이터 셀렉터 신호에 따라 6심볼(symbol) 단위로 제 1부터 제 3 인덱스 디코더(13a,13b,13c)에 분산 전송한다.

<41> 즉, 총 576비트의 코드 심볼 데이터에 대하여 제 1 인덱스 디코더(13a)에는 제 1부터 제 192 비트의 코드 심볼 데이터가, 제 2 인덱스 디코더(13b)에는 제 193부터 제 384 비트의 코드 심볼 데이터가, 제 3 인덱스 디코더(13c)에는 제 385부터 제 576 비트의 코드 심볼 데이터가 분산 전송된다.

<42> 그러면 제 1부터 제 3 인덱스 디코더(13a,13b,13c)에서는 64-ary 직교 변조를 위한 인덱스 디코딩을 실시한다. 이때, 도 6부터 도 9는 제 1 인덱스 디코더(13a)에서 실행되는 디코딩을 설명하기로 한다.

<43> 이때, 코드 심볼 반복기에서 반복되어 전송된 데이터의 메모리 액세스는 실시하지 않는다. 즉, 종래에는 풀 레이트(도 6)를 제외한 하프 레이트(도 7), 쿼터 레이트(도 8) 및 에이트 레이트(도 9)의 데이터는 데이터 처리의 용이성을 위하여 각각 1회/3회/7회의 데이터 반복을 블록 인터리버 메모리부의 이전 블록인 코드 심볼 반복기에서 실시하는데, 본 발명에서는 도 7부터 도 9에 나타낸 바와 같이 코드 심볼 반복 부분(음영(shadow)데이터)에 대하여는 블록 인터리버 메모리부(11)에서 액세스하지 않고 이미 쉬프트 레지스터(12)로 전송된 데이터를 쉬프트 레지스터(12)의 반복 액세스로 해결한다.

<44> 따라서 제 1 인덱스 디코더(13a)에서의 동작은 풀 레이트를 나타낸 도 6에서는 행(row) 어드레스에 따라서 출력 데이터는 1 행(row)부터 16 행(row)까지의 1부터 192의 모든 코드 심볼이 정렬(array)되고, 1 행부터 16행까지의 모든 데이터는 유효한 데이터이다.

<45> 그리고, 하프 레이트를 나타낸 도 7에서는 행(row) 어드레스에 따라서 출력 데이터는 1 행(row), 3, 5, 7, 9, 11, 13 및 15 행(row)까지의 1부터 96의 코드 심볼이 정렬(array)되고, 유효한 데이터는 1 행(row), 3, 5, 7, 9, 11, 13 및 15 행(row)의 데이터가 된다.

<46> 또한, 쿼터 레이트를 나타낸 도 8에서는 행(row) 어드레스에 따라서 출력 데이터는 1 행(row), 5, 9 및 13 행(row)까지의 1부터 64의 코드 심볼이 정렬(array)되고, 유효한 데이터는 1 행(row), 5, 9 및 13 행(row)의 데이터가 된다.

<47> 마지막으로 에이트 레이트를 나타낸 도 9에서는 행(row) 어드레스에 따라서 출력 데이터는 1 행(row) 및 9 행(row)까지의 1부터 24의 코드 심볼이 정렬(array) 되고 유효한 데이터는 1행 및 9행의 데이터가 된다.

<48> 이와 같은 도 7부터 도 9에서는 유효하지 않은 데이터에 대하여는 앞서서도 설명한 바와 같이 음영(shadow) 처리를 하였다.

【발명의 효과】

<49> 이상의 설명에서와 같은 본 발명은 다음과 같은 효과가 있다.

<50> 첫째, 쉬프트 레지스터를 이용하여 데이터를 액세스함으로써 메모리 액세스를 위한 심플 어드레싱 및 데이터 액세스 속도가 향상된다. 따라서 전체 시스템 성능이 향상된다.

<51> 둘째, 쉬프트 레지스터를 이용하여 반복 데이터의 연속적인 액세스를 통한 메모리 액세스 횟수를 줄일 수 있으므로 메모리 액세스 전력이 감소되어, 전체 시스템에서 소요되는 전력이 감소되므로 이통통신 시스템뿐만 아니라 특히 휴대폰에서의 전력 소모를 줄

일 수 있다.

<52> 셋째, 인덱스 디코더를 사용하여 64-ary 직교 변조기의 구동 속도를 향상시키므로
통신 시스템의 전반적인 성능을 향상시킬 수 있다.

【특허청구범위】**【청구항 1】**

복수의 행(row)과, 복수의 열(column) 방향으로 코드 심볼 데이터를 저장하는 블록 인터리버 메모리부와,

상기 블록 인터리버 메모리부에서의 하나의 행(row) 방향의 코드 심볼 데이터를 순차적으로 전송받는 메모리부와,

상기 메모리부의 상기 복수개의 코드 심볼 데이터를 분할하여 수신하고, 수신된 코드 심볼 데이터를 64-ary 직교 변조를 위한 인덱스 디코딩을 실시하는 복수개의 인덱스 디코더와,

상기 블록 인터리버 메모리부로 한번에 하나의 행 방향의 모든 데이터를 전송하도록 제어신호를 발생하고, 상기 메모리부 및 복수개의 인덱스 디코더를 제어하는 제어 신호를 발생시키는 어드레스 발생 및 로직 제어부로 구성됨을 특징으로 하는 인터리버 메모리 제어 장치.

【청구항 2】

제 1 항에 있어서, 상기 n개의 인덱스 디코더는 3개로 구성되고, 상기 메모리부는 쉬프트 레지스터로 구성되는 것을 특징으로 하는 인터리버 메모리 제어 장치.

【청구항 3】

블록 인터리버 메모리부에서 설정된 비트(bit) 수만큼 복수개의 행(row)과, 복수개의 열(column) 방향으로 코드 심볼 데이터를 저장하는 단계와;

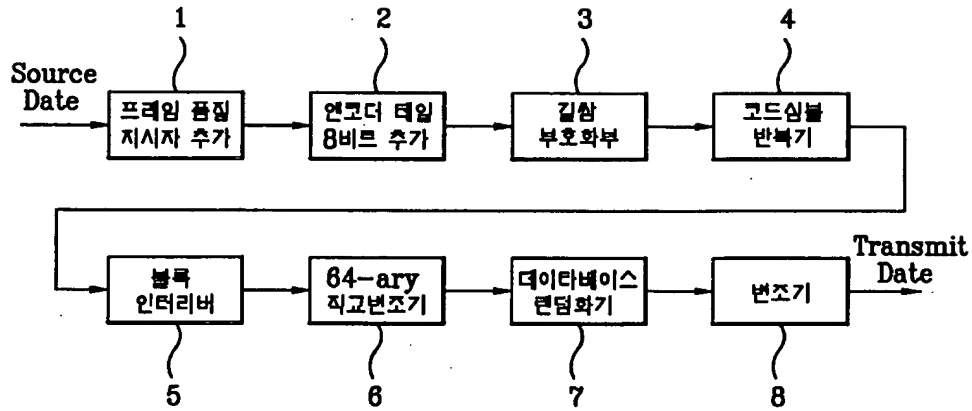
상기 블록 인터리버 메모리부에서 한 클럭(clock)마다 한 개씩의 상기 행(row) 방향의 모든 코드 심볼 데이터를 메모리부로 전송하는 단계와;

상기 메모리부에서 복수개의 인덱스 디코더로 상기 하나의 행 방향의 코드 심볼 데이터를 상기 인덱스 디코더의 개수로 분할하여 전송하는 단계와;

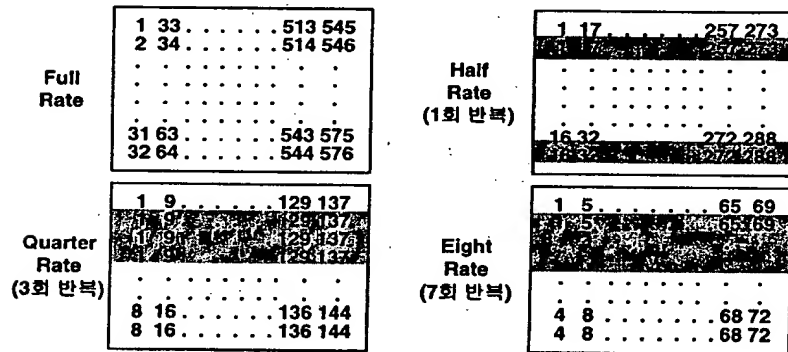
상기 복수개의 인덱스 디코더의 각각에서 상기 전송된 심볼 데이터를 인덱스 디코딩하는 단계로 이루어지는 것을 특징으로 하는 인터리버 메모리 제어 방법.

【도면】

【도 1】



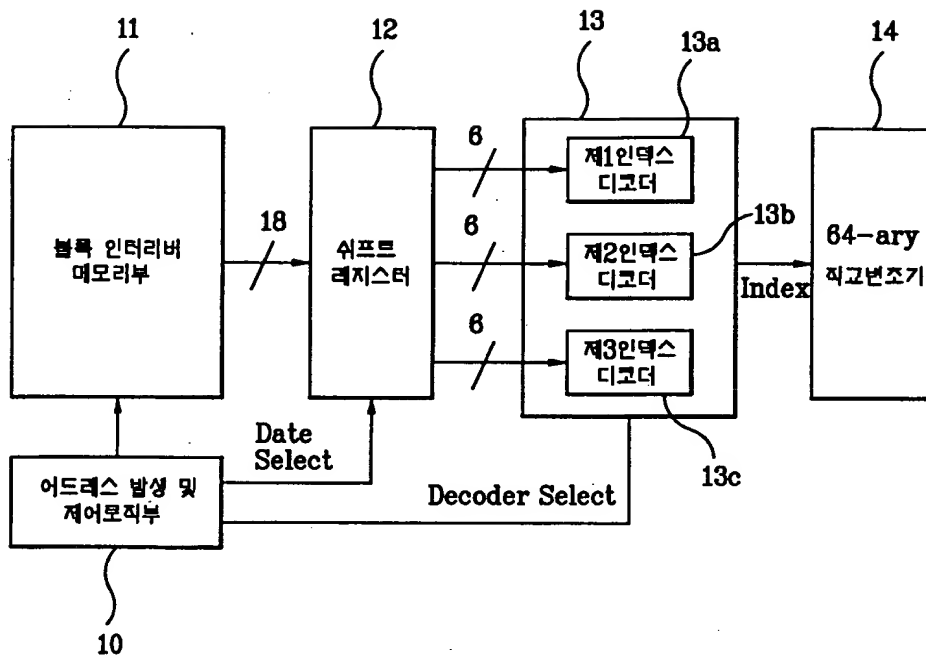
【도 2】



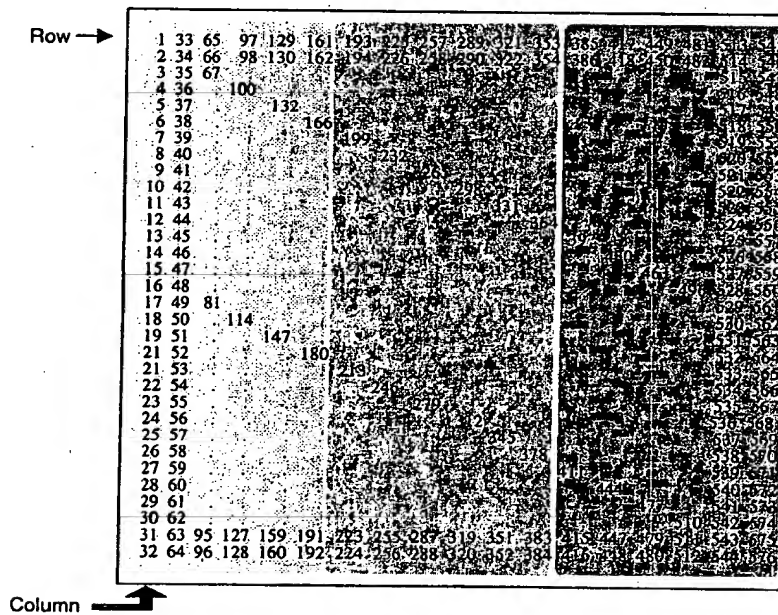
【도 3】

Deinterleaver Array	
Row →	1 17 33 49 65 81 97 113 129 145 161 177 2 18 34 50 66 82 98 114 130 146 162 178 3 19 35 51 67 83 99 115 131 147 163 179 4 20 36 52 68 84 100 116 132 148 164 180 5 21 37 53 69 85 101 117 133 149 165 181 6 22 38 54 70 86 102 118 134 150 166 182 7 23 39 55 71 87 103 119 135 151 167 183 8 24 40 56 72 88 104 120 136 152 168 184 9 25 41 57 73 89 105 121 137 153 169 185 10 26 42 58 74 90 106 122 138 154 170 186 11 27 43 59 75 91 107 123 139 155 171 187 12 28 44 60 76 92 108 124 140 156 172 188 13 29 45 61 77 93 109 125 141 157 173 189 14 30 46 62 78 94 110 126 142 158 174 190 15 31 47 63 79 95 111 127 143 159 175 191 16 32 48 64 80 96 112 128 144 160 176 192
Column →	
Row Address	1 2
Column Address	1 2 3 4 5 6 7 8 9 10 11 12 1 2 3 4 5 6 7 8 9 10 11 12
Output Data	1 17 33 49 65 81 97 113 129 145 161 177 2 18 34 50 66 82 98 114 130 146 162 178
Row Address	3 4
Column Address	1 2 3 4 5 6 7 8 9 10 11 12 1 2 3 4 5 6 7 8 9 10 11 12
Output Data	3 19 35 51 67 83 99 115 131 147 163 179 4 20 36 52 68 84 100 116 132 148 164 180
Row Address	5 6
Column Address	1 2 3 4 5 6 7 8 9 10 11 12 1 2 3 4 5 6 7 8 9 10 11 12
Output Data	5 21 37 53 69 85 101 117 133 149 165 181 6 22 38 54 70 86 102 118 134 150 166 182
Row Address	7 8
Column Address	1 2 3 4 5 6 7 8 9 10 11 12 1 2 3 4 5 6 7 8 9 10 11 12
Output Data	7 23 39 55 71 87 103 119 135 151 167 183 8 24 40 56 72 88 104 120 136 152 168 184
Row Address	9 10
Column Address	1 2 3 4 5 6 7 8 9 10 11 12 1 2 3 4 5 6 7 8 9 10 11 12
Output Data	9 25 41 57 73 89 105 121 137 153 169 185 10 26 42 58 74 90 106 122 138 154 170 186
Row Address	11 12
Column Address	1 2 3 4 5 6 7 8 9 10 11 12 1 2 3 4 5 6 7 8 9 10 11 12
Output Data	11 27 43 59 75 91 107 123 139 155 171 187 12 28 44 60 76 92 108 124 140 156 172 188
Row Address	13 14
Column Address	1 2 3 4 5 6 7 8 9 10 11 12 1 2 3 4 5 6 7 8 9 10 11 12
Output Data	13 29 45 61 77 93 109 125 141 157 173 189 14 30 46 62 78 94 110 126 142 158 174 190
Row Address	15 16
Column Address	1 2 3 4 5 6 7 8 9 10 11 12 1 2 3 4 5 6 7 8 9 10 11 12
Output Data	15 31 47 63 79 95 111 127 143 159 175 191 16 32 48 64 80 96 112 128 144 160 176 192

【도 4】



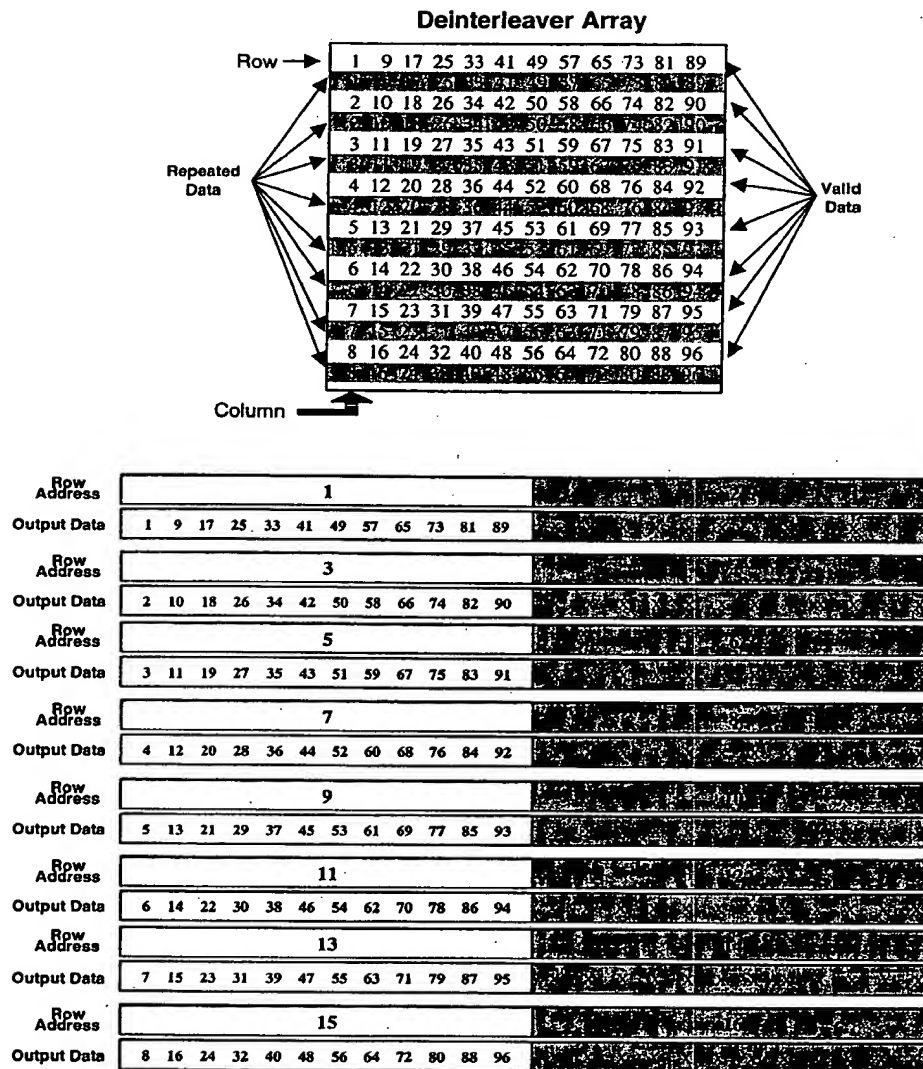
【도 5】



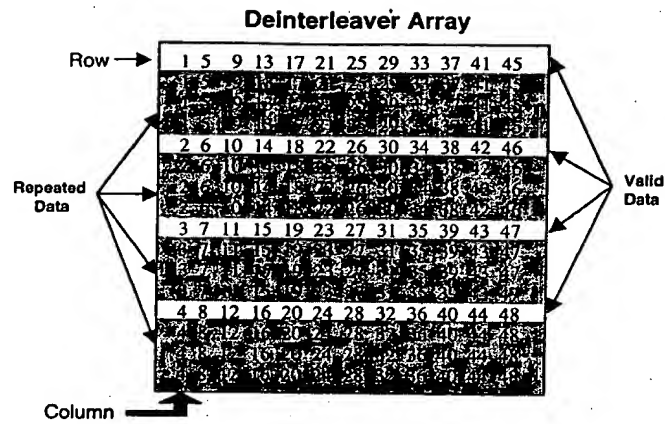
【도 6】

Deinterleaver Array												
Row →	1	17	33	49	65	81	97	113	129	145	161	177
	2	18	34	50	66	82	98	114	130	146	162	178
	3	19	35	51	67	83	99	115	131	147	163	179
	4	20	36	52	68	84	100	116	132	148	164	180
	5	21	37	53	69	85	101	117	133	149	165	181
	6	22	38	54	70	86	102	118	134	150	166	182
	7	23	39	55	71	87	103	119	135	151	167	183
	8	24	40	56	72	88	104	120	136	152	168	184
	9	25	41	57	73	89	105	121	137	153	169	185
	10	26	42	58	74	90	106	122	138	154	170	186
	11	27	43	59	75	91	107	123	139	155	171	187
	12	28	44	60	76	92	108	124	140	156	172	188
	13	29	45	61	77	93	109	125	141	157	173	189
	14	30	46	62	78	94	110	126	142	158	174	190
	15	31	47	63	79	95	111	127	143	159	175	191
	16	32	48	64	80	96	112	128	144	160	176	192
Column →												
Row Address	1						2					
Output Data	1	17	33	49	65	81	97	113	129	145	161	177
Row Address	3						4					
Output Data	3	19	35	51	67	83	99	115	131	147	163	179
Row Address	5						6					
Output Data	5	21	37	53	69	85	101	117	133	149	165	181
Row Address	7						8					
Output Data	7	23	39	55	71	87	103	119	135	151	167	183
Row Address	9						10					
Output Data	9	25	41	57	73	89	105	121	137	153	169	185
Row Address	11						12					
Output Data	11	27	43	59	75	91	107	123	139	155	171	187
Row Address	13						14					
Output Data	13	29	45	61	77	93	109	125	141	157	173	189
Row Address	15						16					
Output Data	15	31	47	63	79	95	111	127	143	159	175	191
	16	32	48	64	80	96	112	128	144	160	176	192

【도 7】



【도 8】



Rpw Address	1	
Output Data	1 5 9 13 17 21 25 29 33 37 41 45 49 53 57 61	
Rpw Address		
Output Data		
Rpw Address	5	
Output Data	2 6 10 14 18 22 26 30 34 38 42 46 50 54 58 62	
Rpw Address		
Output Data		
Rpw Address	9	
Output Data	3 7 11 15 19 23 27 31 35 39 43 47 51 55 59 63	
Rpw Address		
Output Data		
Rpw Address	13	
Output Data	4 8 12 16 20 24 28 32 36 40 44 48 52 56 60 64	
Rpw Address		
Output Data		

【도 9】

